

# Quantifying the Information Leakage in Timing Side Channels in Deterministic Work-Conserving Schedulers (Draft)

Xun Gong, *Student Member, IEEE*, and Negar Kiyavash, *Senior Member, IEEE*

## Abstract

When multiple job processes are served by a single scheduler, the queueing delays of one process are often affected by the others, resulting in a timing side channel that leaks the arrival pattern of one process to the others. In this work, we study such a timing side channel between a regular user and a malicious attacker. Utilizing Shannon's mutual information as a measure of information leakage between the user and attacker, we analyze privacy-preserving behaviors of common work-conserving schedulers. We find that the attacker can always learn perfectly the user's arrival process in a longest-queue-first (LQF) scheduler. When the user's job arrival rate is very low (near zero), first-come-first-serve (FCFS) and round robin schedulers both completely reveal the user's arrival pattern. The near-complete information leakage in the low-rate traffic region is proven to be reduced by half in a work-conserving version of TDMA (WC-TDMA) scheduler, which turns out to be privacy-optimal in the class of deterministic-working-conserving (det-WC) schedulers, according to a universal lower bound on information leakage we derive for all det-WC schedulers.

## I. INTRODUCTION

IT has long been known that event times could be used for covert communication [1]. For instance, by encoding messages in transmission times of events, an event scheduler can create a *timing covert channel* to any observer that sees the time events occur. Some notable timing covert channels include the CPU scheduling channel [2], in which one process encodes a message into sizes of the jobs it hands to a CPU shared with another process which

This work was supported in part by National Science Foundation through the grant CCF 10-65022, CCF 10-54937 CAR, and in part by Air Force through the grant FA9550-11-1-0016, FA9550-10-1-0573.

X. Gong is with the Coordinated Science Laboratory and the Department of Electrical Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (email: xungong1@illinois.edu) N. Kiyavash is with the Coordinated Science Laboratory and the Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (email: kiyavash@illinois.edu)

decodes this information through monitoring CPU's busy period, and the IP timing channel [3], in which messages are embedded in the inter-arrival-times of packets.

More recently, it has been shown that event times also incidentally leak information, resulting in *timing side channels*. Unlike a covert channel, there is no active message sender in a side channel. Instead, an attacker infers information about the other users from the timing evidence left on a shared resource. Such a timing side channel exists between two users sending jobs to the same queue. The queuing delays of one user's jobs convey information about the activities of the other. Timing side channels have been previously exploited to learn the activities of cloud clients and home broadband customers [4], [5]. In cloud computing infrastructures, such as Amazon Elastic Compute Cloud (EC2), a server often hosts jobs from multiple clients. This provides a malicious client with the opportunity to probe workloads of his cloud neighbors [4]. Likewise, a timing side channel can be built in the home digital subscriber line (DSL) router. An attacker ping-ponging the DSL user may learn the user's web traffic pattern because the pings and the user's packets share the downstream queue at the DSL router [5].

In this paper, we study a timing side channel that arises when two users share a joint event queue. One user is assumed to be a malicious attacker who wants to learn the other user's job arrival pattern based on the delays his jobs experience. The amount of coupling between the user and attacker's jobs largely depends on the scheduling policy of the job server. A scheduler certainly can eliminate the side channel by applying a time division multiple access (TDMA) policy, which decouples service to the users but adds unnecessary delays. On the other hand, in work-conserving schedulers, which achieves delay optimality by keeping busy as long as the queue is not empty, timing side channels are inevitable. Kadloor et al. [6] characterized the information leakage of work-conserving schedulers for an attacker that could issue infinitesimally small jobs. This raises the question: *Could side channel information leakage be alleviated if the attacker is not allowed to issue jobs with arbitrarily small sizes?* In fact, in many real systems, there are requirements on acceptable job sizes. For instance, the limit on network packet sizes often pre-fixes [7]. We answer this question by considering a scenario, where users are required to send jobs of comparable sizes. Additionally, we measure the leakage of a scheduler in terms of performance of the best attacker who aims to learn the exact arrival times of the user's jobs. This is a departure from [6], where the attacker's goal was to learn the counts of the user's jobs in each clock period. The current metric captures loss in privacy of the user more accurately. Our main contributions of this work are summarized in the following:

- We develop an information-theoretic framework to analyze timing side channels in job schedulers. Considering a scheduler serving a user and an attacker, we measure the information leakage using Shannon's mutual information between the user's job arrival process and the attacker's job arrival and departure processes.
- We demonstrate that most commonly deployed work-conserving scheduling policies are not privacy optimal: the longest-queue-first (LQF) scheduler leaks the user's arrival pattern completely; when the user's job arrival

rate is near zero, both first-come-first-serve (FCFS) and round robin schedulers completely reveal the user's arrival pattern, while a work-conserving TDMA-like scheduler leaks the user's arrival process half of the time.

- We derive a lower-bound on information leakage for all deterministic work-conserving (det-WC) schedulers, where the server takes deterministic actions and stays busy as long as there are jobs to serve. The lower bound shows that in the low-rate traffic region, the attacker learns the user's arrival pattern for at least half of the time. The implication of this study is that deploying det-WC schedulers in applications, in which privacy is a concern, is a poor choice.

## II. RELATED WORK

Traditionally, timing channels are for the most part studied in the context of covert communication. Most of the literature focuses on the capacity of such channels. Anantharam and Verdú [8] studied the timing channel between the arrival and departure process of a single user  $M/G/1$  queue, and showed that capacity is minimized when the service times of jobs are exponentially distributed. For such a queue, bounds on the capacity for Bounded Service Timing Channels (BSTC), in which the service time distributions have bounded support, were derived in [9]. Riedl et. al [10] considered the usage of the same channel with finite-length codewords, and obtained a lower bound on the maximal rate achievable. A covert channel between two job processes sharing a round robin scheduler was studied in [2]. Assuming all jobs have the same size, it was proved that the channel capacity is  $\log\left(\frac{1+\sqrt{5}}{2}\right)$  bits per time slot. Strategies for mitigating timing covert channels were studied in [11]–[14]. The main proposed countermeasure idea is to weaken the correlation between event times seen by the sender and receiver via injecting ‘dummy’ delays.

On the application side, timing side channels were exploited in network traffic analysis to compromise user anonymity. In [15], round-trip times (RTTs) of probe packets sent to routers were measured to estimate available bandwidths at the router, which were subsequently used to expose the identity of relays participating in a circuit of the anonymous communication networks, such as Tor [16] or MorphMix [17]. In [18], Kiyavash et al. designed and implemented a spyware communication circuit in the widely used carrier sense multiple access with collision avoidance (CSMA/CA) protocol, using the timing channel resulting from transmission of packets. In [5] and [19], it was shown that an attacker can create a timing side channel inside a DSL router using frequent pings, and recover DSL user's traffic pattern from monitored RTTs. Queuing side channels in shared queues were analyzed in [6] and [20], where the information leakage was measured by minimum-mean-square-error and equivocation, respectively. With the goal of measuring the number of jobs from the user in a clock period, it was shown in both [6] and [20] that an FCFS scheduler completely leaked the user's traffic pattern if the attacker could send at least one job in every clock period. Additionally, assuming the attacker was able to issue jobs with infinitesimal

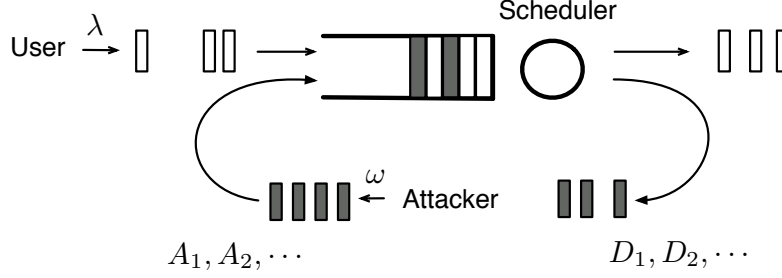


Fig. 1. A scheduler services jobs from two arrival processes; one from a malicious attacker (solid) and one from a regular user (blank). The attacker sends jobs to the scheduler to sample the queue status, based on which he learns about the arrival pattern of the other user.

sizes, it was proven in [6] that round robin is privacy optimal among all work-conserving schedulers; yet it leaks substantial information about the user.

### III. PROBLEM FORMULATION

In this section, we introduce the notation and system model. Throughout this section: bold script  $\mathbf{A}$  denotes the infinite sequence  $\{A_1, A_2, \dots\}$ ,  $\mathbf{A}^n$  denotes the finite sequence  $\{A_1, A_2, \dots, A_n\}$ , and  $\mathbf{A}_i^j$  denotes the subsequence  $\{A_i, A_{i+1}, \dots, A_j\}$ , where  $j \geq i$ .

#### A. System Model

We consider the timing side channel in a scheduler processing jobs from a regular user and a malicious attacker in discrete time, as depicted by Figure 1. In each time slot, the user (and the attacker) either issues one job or remains idle. All jobs have the same size and take one time slot to service. The user sends jobs according to a Bernoulli process with rate  $\lambda$ . The attacker, who wants to infer the user's arrival times, picks time slots according to his attack strategy and sends jobs with the long term rate  $\omega$ , not exceeding  $1 - \lambda$ , in order to preserve the queue's stability. We assume all arrival and departure events occur at the beginning of time slots.

#### B. Information Leakage Metric

We measure information leakage in this timing side channel by Shannon's mutual information between the user's arrival process and the attacker's observations, comprised of his arrival and departure times. Similar metrics, e.g., Shannon's equivocation, have been frequently used for quantifying information leakage in communication systems, such as the wiretap channel [21]. Denote the arrival event sequence of the user's jobs in each time slot by  $\boldsymbol{\delta} = \{\delta_1, \delta_2, \dots\}$ , where  $\delta_i \sim \text{Bernoulli}(\lambda)$ ,  $i \in \mathbb{Z}$ , and denote the arrival and departure times of the attacker's jobs by  $\mathbf{A} = \{A_1, A_2, \dots\}$  and  $\mathbf{D} = \{D_1, D_2, \dots\}$ , respectively.

*Definition 1:* The information leakage of a timing side channel in a queue shared by a user and an attacker is defined as

$$\mathcal{L}(\lambda) = \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1 - \lambda} \lim_{n \rightarrow \infty} \frac{I(\delta^n; \mathbf{A}^m, \mathbf{D}^m)}{n}, \quad (1)$$

where  $I(\cdot; \cdot)$  denotes Shannon's mutual information,  $\lambda$  is the user's arrival rate, and  $m$  is the number of jobs the attacker has issued by time  $n$ :

$$m = \sup\{k : A_k \leq n\}. \quad (2)$$

The leakage  $\mathcal{L}$  characterizes the information gain of the attacker deploying the best possible attack strategy satisfying the rate restriction. Hence, a larger leakage  $\mathcal{L}$  signifies a larger compromise in the user's privacy through the timing side channel. Let  $H(\lambda)$  denote the entropy rate of the user's arrival process, which is assumed to be Bernoulli.

*Definition 2:* The information leakage ratio of a timing side channel in a queue shared by a user and an attacker is defined as

$$\mathcal{R}(\lambda) = \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1 - \lambda} \lim_{n \rightarrow \infty} \frac{I(\delta^n; \mathbf{A}^m, \mathbf{D}^m)}{nH(\lambda)}. \quad (3)$$

The information leakage ratio  $\mathcal{R}$  is a better metric for comparing the leakage across users with various rates. The value of  $\mathcal{L}$  (and  $\mathcal{R}$ ) clearly depends on the scheduling policy. For instance, for the TDMA policy, in which fixed time slots are preassigned for serving each arrival process, both  $\mathcal{L}$  and  $\mathcal{R}$  are zero. This is because service times of the attacker's jobs are statistically independent of the user's arrival pattern. Unfortunately, TDMA is wasteful and adds significant delays by causing the scheduler to idle. Therefore, such complete isolation of users' job processes is often not desired in practice. In this work, we analyze the information leakage of timing channels in work-conserving (delay-optimal) schedulers and investigate whether good policies that are simultaneously privacy and delay optimal exist.

#### IV. INFORMATION LEAKAGE IN DETERMINISTIC WORK-CONSERVING SCHEDULERS

In this section, we characterize or derive bounds on the leakage in the class of deterministic-work-conserving (det-WC) schedulers. These schedulers service jobs in a deterministic fashion and do not idle as long as there are jobs in the queue. Our main results are summarized in Figure 2.

We show that even when the attacker is required to send jobs of a comparable size to the user, all det-WC schedulers leak at least half of the user's traffic pattern in the low-rate region. This is proved by deriving a universal lower bound for all det-WC schedulers as depicted in Figure 2.

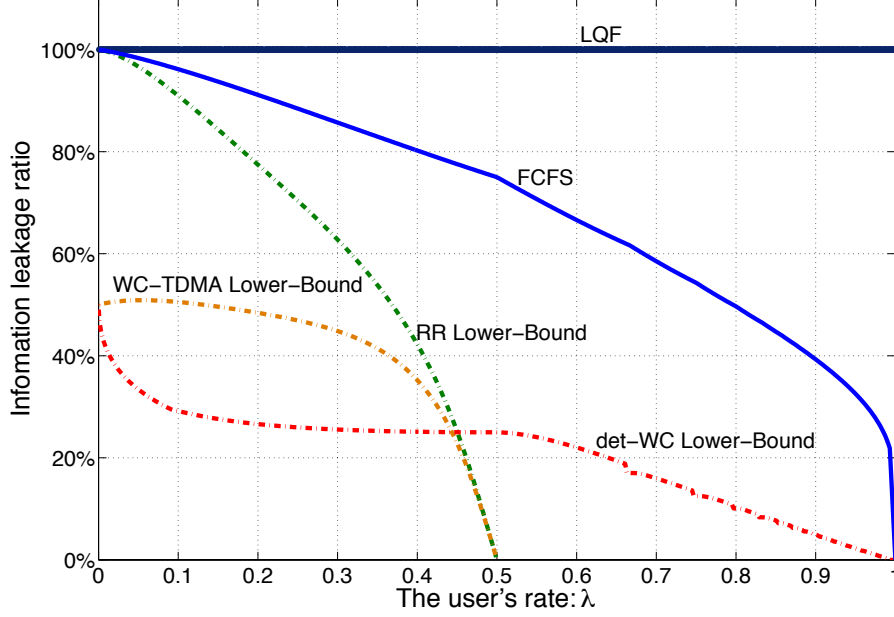


Fig. 2. Information leakage ratios in deterministic-work-conserving schedulers. In LQF, the user's arrival process is completely leaked to the attacker, which also occurs in the FCFS and round robin scheduler when the user's rate is very low. A work-conserving TDMA scheduler reduces the fraction of leaked information in the low-rate region by half; the lower-bound on the leakage of WC-TDMA is tight at  $\lambda \rightarrow 0$ .

The attacker learns completely the arrival process of the user in an LQF scheduler by simply maintaining his own queue length at one (the flat solid line at the top in Figure 2). In an FCFS scheduler, instead of the exact arrival times, the attacker can infer the number of jobs (arrivals) of the user between any of his two consecutive jobs by sampling the queue at his maximum rate. This leads to a severe leakage in the arrival process of a user with low arrival rate (the blue solid curve in Figure 2), as the attacker can sample the queue frequently enough to obtain an accurate estimate of the user's arrival event in each time slot. We derive a lower bound on the privacy leakage for round robin (the green dashed curve in Figure 2), which can be achieved by an attacker who issues a new job right after his previous job is serviced. This attack strategy allows the attacker to detect when the user's queue gets empty. As depicted in Figure 2, it provides sufficient information about the user's arrival pattern at the low-rate region, resulting in a complete information leakage when the user's arrival rate  $\lambda \rightarrow 0$ .

The near-complete information leakage in the low-rate region is alleviated by the work-conserving TDMA (WC-TDMA) scheduler. Like TDMA, the WC-TDMA scheduler reserves slots for each arrival process; e.g., odd slots for the user and even ones for the attacker. However, in each time slot, if the preassigned user has no jobs, the scheduler serves the other user with jobs waiting for service. Such work-conserving behavior enables the attacker to correctly detect arrivals on time slots reserved by the user. We derive a lower bound on the leakage of a WC-TDMA scheduler, which is proved to be tight when the user's rate  $\lambda \rightarrow 0$  (the orange dashed curve in Figure 2). This means the attacker can learn the arrival pattern of a low-rate user perfectly for half of the time, and further implies

that for  $\lambda \rightarrow 0$ , WC-TDMA is a privacy optimal policy in the det-WC class as it meets the det-WC universal lower-bound.

#### A. Longest-Queue-First

We first analyze the leakage of an LQF scheduler, which we can exactly characterize. In each time slot, the LQF scheduler services the first buffered job from the user that has more jobs queued up so far. In the case of a tie, the scheduler serves a predetermined user first.

Since the LQF scheduler takes actions by comparing queue lengths of users, a smart attacker can accurately learn the change in the user's queue state by maintaining his queue length constantly at one. Assuming the user has priority of service at a tie, such an attacker always knows whenever the user's queue size passes 0 and detects every job sent by the user, as further explained below.

*Theorem 4.1:* The information leakage of an LQF scheduler serving a user and an attacker is given by

$$\mathcal{L}_{LQF}(\lambda) = H(\lambda), \quad (4)$$

where  $\lambda$  is the user's arrival rate. Consequently,  $\mathcal{R}_{LQF}(\lambda) = 1$ , for all  $0 < \lambda < 1$ .

*Proof:* Consider a *nonstop monitoring* attack strategy (Figure 3), where the attacker issues a new job immediately after his previous is serviced. Recall in our model, all arrival and departure events happen at the beginning of time slots. Thus, in the nonstop monitoring attack, we have

$$A_k = D_{k-1}, \quad k \in \mathbb{Z}. \quad (5)$$

Such an attacker always has a single job in the queue. Assume the user gets served first when a tie happens. Then, the scheduler never serves the attacker unless the user has no job left. As a result, whenever the user issues a new job, the attacker experiences a time slot of delay, i.e.,

$$\delta_i = \begin{cases} 0 & \text{if } \exists k \in \mathbb{Z}, \text{ s.t. } D_k = i + 1 \\ 1 & \text{otherwise} \end{cases}, \quad i \in \mathbb{Z}. \quad (6)$$

Similarly, if the attacker has priority of service when there is a tie in queue lengths, he would get served if and only if the user's queue length falls below 2, in which case (6) also holds.

Therefore, we can obtain a lower bound on information leakage which results from the nonstop-monitoring attack

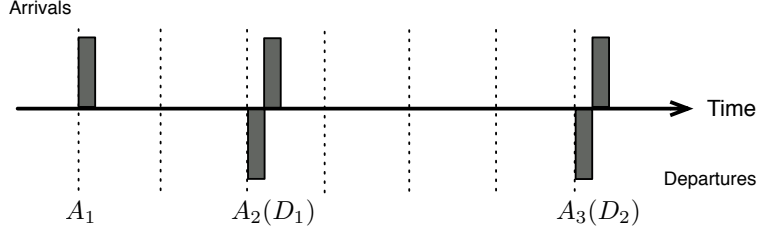


Fig. 3. Nonstop monitoring: the attacker issues a new job right after a previous job departs from the queue.

as follows:

$$\begin{aligned}
 \mathcal{L}_{LQF}(\lambda) &\geq \lim_{n \rightarrow \infty} \frac{I(\delta^n; \mathbf{A}^m, \mathbf{D}^m)}{n} \\
 &\stackrel{(a)}{=} \lim_{n \rightarrow \infty} \frac{I(\delta^n; \mathbf{D}^m)}{n} \\
 &= H(\lambda) - \lim_{n \rightarrow \infty} \frac{H(\delta^n | \mathbf{D}^m)}{n} \\
 &\stackrel{(b)}{=} H(\lambda).
 \end{aligned} \tag{7}$$

where (a) and (b) follow from (5) and (6), respectively. Additionally, since the leakage is always upper-bounded by the total entropy rate of the user's arrival process,  $H(\lambda)$ , we have  $\mathcal{L}_{LQF}(\lambda) = H(\lambda)$ . ■

LQF is a low-complexity approximation of the MaxWeight scheduling, a throughput-optimal algorithm applied in network switches [22]. It requires less buffer storage than other common scheduling algorithms, such as FCFS and round robin [23]. However, as seen in Theorem 4.1, LQF fully exposes arrival pattern of the user to an attacker.

### B. First-Come-First-Serve

We subsequently analyze the leakage of FCFS, a simple service policy widely applied in network systems. At each time slot, the FCFS scheduler services the job at the head of the queue.<sup>1</sup> FCFS reveals the queue length  $q(\cdot)$  of the buffer to an attacker through queueing delays of his jobs because

$$q(A_k) = D_k - A_k - 1, \quad k \in \mathbb{Z}, \tag{8}$$

where '1' accounts for the service time of the  $k^{th}$  attacker's job. As a result, the attacker can frequently sample the state of the buffer queue and then estimate the number of the user's arrivals.

Let  $N(t)$  denote the counting function associated with the user's arrivals at time  $t$ , then

$$N(t) = \sum_{j=1}^t \delta_j, \quad t \in \mathbb{Z}. \tag{9}$$

<sup>1</sup>For the sake of convenience, we assume that when both the user and the attacker issue a job in one time slot, the attacker's job enters the queue first.



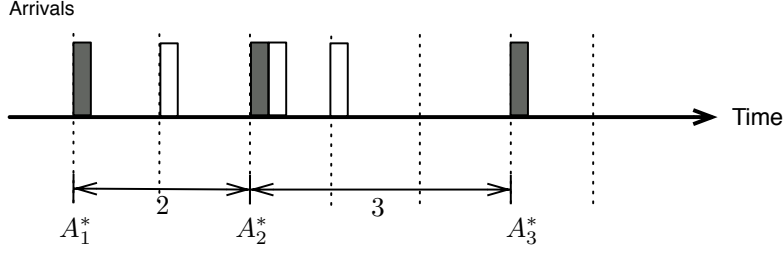


Fig. 4. Periodic sampling: given a sampling rate  $\omega$ , the attacker issues jobs (solid) periodically, with inter-arrival times chosen from  $\lfloor \frac{1}{\omega} \rfloor$  and  $\lceil \frac{1}{\omega} \rceil$ . For example, if  $\omega = 0.4$ , the inter-arrival time  $A_i^* - A_{i-1}^*$  would take value of 2 and 3 with equal probability.

The following theorem on optimal sampling of a Bernoulli processes is necessary for proving our main result on information leakage of the FCFS scheduler. For the ease of notation, we define in the following

$$\alpha_\epsilon \triangleq \frac{\lceil \frac{1}{\epsilon} \rceil - \frac{1}{\epsilon}}{\lceil \frac{1}{\epsilon} \rceil - \lfloor \frac{1}{\epsilon} \rfloor}, \quad \forall 0 < \epsilon < 1. \quad (10)$$

*Theorem 4.2:* Consider sampling a Bernoulli arrival process  $\delta = \{\delta_1, \delta_2, \dots\}$  at times  $\mathbf{A} = \{A_1, A_2, \dots\}$ . For a fixed sampling rate  $\omega = \lim_{k \rightarrow \infty} \frac{k}{A_k}$ , the following periodic sampling strategy (Figure 4) is optimal:

$$A_k^* - A_{k-1}^* = \begin{cases} \lfloor \frac{1}{\omega} \rfloor & \text{w.p. } \alpha_\omega \\ \lceil \frac{1}{\omega} \rceil & \text{w.p. } 1 - \alpha_\omega \end{cases}, \quad k \in \mathbb{Z}. \quad (11)$$

The optimality is defined in the sense of maximizing the entropy rate of the sampled process. This optimal value is given by

$$\lim_{k \rightarrow \infty} \frac{H(N(A_1^*), \dots, N(A_k^*))}{k} = \alpha_\omega H\left(\sum_{i=1}^{\lfloor \frac{1}{\omega} \rfloor} \delta_i\right) + (1 - \alpha_\omega) H\left(\sum_{i=1}^{\lceil \frac{1}{\omega} \rceil} \delta_i\right). \quad (12)$$

*Proof:* See Appendix A. ■

*Theorem 4.3:* The information leakage of an FCFS scheduler serving a user and an attacker is given by

$$\mathcal{L}_{FCFS}(\lambda) = (1 - \lambda) \left( \alpha_{1-\lambda} H\left(\sum_{i=1}^{\lfloor \frac{1}{1-\lambda} \rfloor} \delta_i\right) + (1 - \alpha_{1-\lambda}) H\left(\sum_{i=1}^{\lceil \frac{1}{1-\lambda} \rceil} \delta_i\right) \right), \quad (13)$$

where  $\lambda$  is the user's arrival rate, and  $\delta_i$ 's are i.i.d.  $Bernoulli(\lambda)$  random variables. If  $\frac{1}{1-\lambda} \in \mathbb{Z}$ , (13) is simplified to  $\mathcal{L}_{FCFS}(\lambda) = (1 - \lambda) H\left(\sum_{i=1}^{\frac{1}{1-\lambda}} \delta_i\right)$ .

In particular, in the low-rate region, the user's arrival pattern is completely leaked to the attacker:

$$\lim_{\lambda \rightarrow 0} \mathcal{R}_{FCFS}(\lambda) = 1. \quad (14)$$

*Proof:* We first prove the *converse* part of (13) by showing that there exists no attack strategy that allows the attacker to learn more information than (13).

From (1) and (8), we have

$$\begin{aligned}
\mathcal{L}_{FCFS}(\lambda) &= \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1-\lambda} \lim_{n \rightarrow \infty} \frac{I(\delta^n; \mathbf{A}^m, q(A_1), q(A_2), \dots, q(A_m))}{n} \\
&\stackrel{(a)}{\leq} \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1-\lambda} \lim_{n \rightarrow \infty} \frac{I(\delta^n; \mathbf{A}^m, \mathbf{X}^m)}{n} \\
&= \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1-\lambda} \lim_{n \rightarrow \infty} \frac{H(\mathbf{X}^m) + H(\mathbf{A}^m | \mathbf{X}^m) - H(\mathbf{A}^m | \delta^n) - H(\mathbf{X}^m | \mathbf{A}^m, \delta^n)}{n} \\
&\stackrel{(b)}{=} \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1-\lambda} \lim_{n \rightarrow \infty} \frac{H(\mathbf{X}^m) + H(\mathbf{A}^m | \mathbf{X}^m) - H(\mathbf{A}^m | \delta^n)}{n} \\
&\stackrel{(c)}{\leq} \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1-\lambda} \lim_{n \rightarrow \infty} \frac{H(\mathbf{X}^m)}{n}
\end{aligned} \tag{15}$$

where  $X_k$  is the number of the user's jobs that have arrived between time  $A_{k-1}$  and  $A_k$  and  $X_k = \sum_{j=A_{k-1}}^{A_k-1} \delta_j$ .

(a) results from the application of data processing inequality [24, Theorem 2.8.1] to the Markov chain

$$\delta^n \rightarrow \mathbf{A}^m, \mathbf{X}^m \rightarrow \mathbf{A}^m, q(A_1), \dots, q(A_m). \tag{16}$$

(b) follows from the fact that  $\mathbf{X}^m$  is a deterministic function of  $\mathbf{A}^m$  and  $\delta^n$ , and (c) results from the Markov chain<sup>2</sup>

$$\delta^n \rightarrow \mathbf{X}^m \rightarrow \mathbf{A}^m. \tag{17}$$

Recall the counting function in (9). The number of user's jobs sent by time  $A_k$  is  $N(A_k) = \sum_{j=1}^k X_j$ . Hence, (15) can be rewritten as

$$\mathcal{L}_{FCFS}(\lambda) \leq \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1-\lambda} \lim_{n \rightarrow \infty} \frac{H(N(A_1), \dots, N(A_m))}{n}. \tag{18}$$

This implies that the attacker learns at most a sampled version of the user's arrival process through this side channel.

From (2), (18) can be rewritten as

$$\begin{aligned}
\mathcal{L}_{FCFS}(\lambda) &\leq \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1-\lambda} \lim_{n \rightarrow \infty} \frac{H(N(A_1), \dots, N(A_m))}{m} \cdot \frac{m}{A_m} \cdot \frac{A_m}{n} \\
&= \max_{\mathbf{A}: \lim_{k \rightarrow \infty} \frac{k}{A_k} < 1-\lambda} \omega \lim_{n \rightarrow \infty} \frac{H(N(A_1), \dots, N(A_m))}{m}.
\end{aligned} \tag{19}$$

<sup>2</sup>For FCFS, attack strategies can be divided into two types. The first type is fully independent with the user's behavior. The second type makes use of past departure history;  $A_k$  depends on previous departure  $D_{k-1}$ . Since  $D_{k-1}$  is uniquely determined once  $\mathbf{X}^{k-1}$  is given,  $\mathbf{A}^m$  must be independent with  $\delta^n$  given  $\mathbf{X}^m$ . This implies the Markov chain in (16).

Applying Theorem 4.2,

$$\begin{aligned} \mathcal{L}_{FCFS}(\lambda) &\leq \max_{\omega < 1-\lambda} \omega \left( \alpha_{\omega} H \left( \sum_{i=1}^{\lfloor \frac{1}{\omega} \rfloor} \delta_i \right) + (1 - \alpha_{\omega}) H \left( \sum_{i=1}^{\lceil \frac{1}{\omega} \rceil} \delta_i \right) \right) \\ &\stackrel{(d)}{=} (1 - \lambda) \left( \alpha_{1-\lambda} H \left( \sum_{i=1}^{\lfloor \frac{1}{1-\lambda} \rfloor} \delta_i \right) + (1 - \alpha_{1-\lambda}) H \left( \sum_{i=1}^{\lceil \frac{1}{1-\lambda} \rceil} \delta_i \right) \right), \end{aligned} \quad (20)$$

where (d) simply applies the monotonic-increasing property of  $H \left( \sum_{i=1}^k \delta_i \right)$  as a function of  $k$ . This completes the proof for the converse.

To prove the achievability of (13), we consider the periodic sampling strategy defined in (11), and derive a lower bound on the information leakage, which turns out to meet the upper bound in (13). See Appendix B.

Once (13) is proven, we take the limit of leakage ratio as  $\lambda \rightarrow 0$ ,

$$\lim_{\lambda \rightarrow 0} \mathcal{R}_{FCFS}(\lambda) \stackrel{(e)}{=} \frac{H(\delta_1)}{H(\lambda)} \stackrel{(f)}{=} 1. \quad (21)$$

where (e) holds because  $\lim_{\lambda \rightarrow 0} \alpha_{1-\lambda} = 0$  according to (10) and (f) follows from the Bernoulli distribution of  $\delta_i$ 's. ■

Theorem 4.3 proves that the attacker can recover the number of user's jobs arriving in each sampling period, which becomes an accurate estimate of the user's job arrival pattern if the sampling frequency is high. When the user sends jobs at a very low rate, the attacker can sample the queue state almost every time slot, and thus would learn the user's arrival pattern completely (See Figure 2).

### C. Round Robin

The next policy we study is round robin, where two users take turns to receive services. In each time slot, the service is switched to the next user who has jobs waiting in the queue; the scheduler never serves any single user continuously unless the other user runs out of jobs. To derive a lower bound on the information leakage, we consider the nonstop monitoring attack introduced in §IV-A (Figure 3), where the job arrival times and departure times satisfy (5). For round robin, this attack forces the scheduler to serve the attacker continuously if possible. As a result, the attacker learns when the user's queue becomes empty, as illustrated in Figure 5, or

$$q(A_k) \begin{cases} = 0 & \text{if } D_k - A_k = 1 \\ > 0 & \text{if } D_k - A_k = 2 \end{cases}, \quad k \in \mathbb{Z}. \quad (22)$$

Notice in (22), the time gap between two consecutive departures of attacker is at most two time slots. Hence, this attack is only applicable for the region, where the user's rate  $\lambda \leq 0.5$ .

Define the *busy period* of the system as the time gap between two times when the attacker finds the user's queue

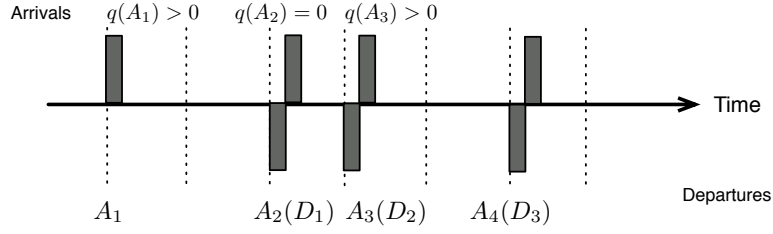


Fig. 5. Nonstop monitoring on the round-robin scheduler: the attacker's job are serviced instantly if there is no job from the user in the queue. Otherwise the scheduler needs to serve the user for one time slot before switching back to the attacker. As a result, the queuing delay  $D_k - A_k$  indicates whether user's queue is empty, i.e.,  $q(A_k) = 0$ .

is empty, and denote the  $r^{th}$  busy period by  $B_r$ .  $B_r$ 's can be written as

$$B_r = \inf \left\{ A_k : A_k > \sum_{j=1}^{r-1} B_j \text{ and } q(A_k) = 0 \right\} - \sum_{j=1}^{r-1} B_j. \quad (23)$$

What the attacker learns from the side channel is summarized by the busy period sequence  $\mathbf{B} = \{B_1, B_2, \dots\}$ .

*Theorem 4.4:* The information leakage of a round robin scheduler serving a user and an attacker is lower-bounded by

$$\mathcal{L}_{RR}(\lambda) \geq (1 - 2\lambda) H(B), \quad \text{for } \lambda \leq 0.5, \quad (24)$$

where  $\lambda$  is the user's arrival rate,  $B$  is the random variable distributed as

$$\mathbb{P}(B = k) = \begin{cases} 2^{r-1} \lambda^{r-1} (1 - \lambda)^r \max \left\{ \sum_{j=1}^{\lfloor \frac{r-2}{2} \rfloor} \frac{(r-2)!}{(r-2-2j)! j! (j+1)!} 2^{-2j-1}, 1 \right\} & \text{if } k = 2r - 1, \\ 0 & \text{otherwise,} \end{cases} \quad (25)$$

where  $r \in \mathbb{Z}^+$ . In particular, when the user's rate is very low, the attacker learns completely the user's traffic pattern, i.e.,

$$\lim_{\lambda \rightarrow 0} \mathcal{R}_{RR}(\lambda) = 1. \quad (26)$$

*Proof:* Similar to (7), the nonstop monitoring attack results in a lower bound on the leakage given by

$$\begin{aligned} \mathcal{L}_{RR}(\lambda) &\geq \lim_{n \rightarrow \infty} \frac{I(\boldsymbol{\delta}^n; \mathbf{D}^m)}{n} \\ &\stackrel{(a)}{=} \lim_{n \rightarrow \infty} \frac{H(\mathbf{D}^m)}{n} \end{aligned} \quad (27)$$

where (a) holds because the attacker's departure times for the round robin scheduler are deterministic once the

user's arrival pattern is known. From (5) and (22), we have

$$\begin{aligned} H(\mathbf{D}^m) &= H(\mathbb{1}_{\{q(A_1)=0\}}, \mathbb{1}_{\{q(A_2)=0\}}, \dots, \mathbb{1}_{\{q(A_m)=0\}}) \\ &\stackrel{(b)}{\geq} H(B_1, B_2, \dots, B_m) \end{aligned} \quad (28)$$

where (b) follows from the definition of busy periods in (23).

It can be further shown that the busy periods seen by the attacker,  $B_r$ 's, are i.i.d. distributed as (25) and have mean  $\frac{1}{1-2\lambda}$ . The proof is presented in Appendix E. Therefore, plugging (28) into (27) proves (24). Additionally, taking the limit of (24) at  $\lambda \rightarrow 0$ , the leakage ratio is lower-bounded as

$$\begin{aligned} \lim_{\lambda \rightarrow 0} \mathcal{R}_{RR}(\lambda) &\geq \lim_{\lambda \rightarrow 0} (1-2\lambda) \frac{H(B)}{H(\lambda)} \\ &\stackrel{(c)}{\geq} \lim_{\lambda \rightarrow 0} \frac{-(1-\lambda) \log(1-\lambda) - \lambda(1-\lambda)^2 \log(\lambda(1-\lambda)^2)}{-(1-\lambda) \log(1-\lambda) - \lambda \log \lambda}, \end{aligned} \quad (29)$$

where (c) holds by plugging in the PMF of random variable  $B$  in (25) only for the terms  $k=1$  and  $k=3$ . The limit on the right hand side of inequality (c) goes to 1 as  $\lambda \rightarrow 0$ . This completes the proof. ■

Round robin is one of the simplest scheduling algorithms in multi-processor operation systems and known for fairness [25]. However, our analysis shows that in the low-rate region the round robin scheduler almost entirely leaks a user's traffic pattern through the timing side channel (See Figure 2).

#### D. Work-Conserving TDMA

The schedulers analyzed so far all leak substantial information about the user's traffic, especially when the user's job arrival rate is low. This imposes serious threat to user privacy since many network systems have light workloads. In fact, studies have shown that average server utilization in real world data centers is only about 5% to 20% [26]. In this section, we study WC-TDMA, a tweak of TDMA, which can reduce the information leakage in the low-rate traffic region by half.

In WC-TDMA, time slots are pre-assigned to the user and attacker equally. Unlike TDMA, if in some slot the reserved user has no jobs left, the WC-TDMA scheduler serves the other user who has jobs waiting in the queue. For the sake of convenience, we assume all odd time slots are assigned to the user, and all even slots are assigned to the attacker.

We consider an attack strategy, where the attacker sends a job in each odd slot and stays idle in all even slots, i.e.,

$$A_k = 2k - 1, \quad k \in \mathbb{Z}. \quad (30)$$

Clearly, this attack consumes half of the service capacity, so is only applicable when the user's rate  $\lambda \leq 0.5$ .

Since odd time slots are reserved for the user, the attacker's jobs sent on those slots would experience delays as

follows:

$$D_k - A_k = \begin{cases} 1 & \text{if } q(A_k) = 0 \text{ and } \delta_{A_k} = 0 \\ 2 & \text{otherwise} \end{cases}, \quad k \in \mathbb{Z}. \quad (31)$$

Therefore, the attacker learns the time slots when the queue is empty and the user has not issued a job.

Define the ‘busy period’,  $B'_r, r \in \mathbb{Z}$ , to be the time gap between two successive times when the attacker sees an empty queue. Then,

$$B'_r = \inf \left\{ A_k : A_k > \sum_{j=1}^{r-1} B'_j \text{ and } q(A_k) = 0 \right\} - \sum_{j=1}^{r-1} B'_j. \quad (32)$$

*Theorem 4.5:* The information leakage ratio of a WC-TDMA scheduler serving a user and an attacker is lower-bounded by

$$\mathcal{L}_{WC-TDMA}(\lambda) \geq \frac{1-2\lambda}{2-2\lambda} H(B), \quad \text{for } \lambda \leq 0.5, \quad (33)$$

where  $\lambda$  is the user’s arrival rate, and  $B$  is the random variable distributed as (25).

*Proof:* Like (27) in the proof of Theorem 4.4, a lower bound on information leakage can be written as

$$\begin{aligned} \mathcal{L}_{WC-TDMA}(\lambda) &\geq \lim_{k \rightarrow \infty} \frac{H(\mathbf{D}^k)}{2k} \\ &\stackrel{(a)}{=} \lim_{k \rightarrow \infty} \frac{H(\mathbb{1}_{\{q(1)=0\}}, \mathbb{1}_{\{q(3)=0\}}, \dots, \mathbb{1}_{\{q(2k-1)=0\}})}{2k} \\ &\stackrel{(b)}{\geq} \lim_{k \rightarrow \infty} \frac{H(B'_1, B'_2, \dots, B'_k)}{2k} \end{aligned} \quad (34)$$

where (a) follows from (30) and (31), and (b) follows from (32).

It can be shown that the busy periods seen by the attacker,  $B'_r$ ’s, are i.i.d. distributed as (25) and have mean  $\frac{2-2\lambda}{1-2\lambda}$  (See Appendix F). Hence, (34) readily implies the desired lower bound. ■

*Corollary 4.6:* The information leakage of a WC-TDMA scheduler serving a user and an attacker is given by

$$\lim_{\lambda \rightarrow 0} \mathcal{R}_{WC-TDMA}(\lambda) = \frac{1}{2}, \quad \text{if } \lambda \rightarrow 0. \quad (35)$$

*Proof:* Taking the limit of (34) at  $\lambda \rightarrow 0$ , we get

$$\lim_{\lambda \rightarrow 0} \mathcal{R}_{WC-TDMA}(\lambda) \geq \lim_{\lambda \rightarrow 0} \frac{1-2\lambda}{2-2\lambda} \frac{H(B)}{H(\lambda)} \stackrel{(a)}{\geq} \frac{1}{2}, \quad (36)$$

where (a) follows from (29).

We subsequently derive an upper-bound on the leakage ratio. Define  $S_k$  to be the earliest time when the  $k^{th}$

attacker's job can receive service. Then  $S_k = \max\{D_{k-1}, A_k\}$ , and we have

$$\begin{aligned}
I(\delta^n; \mathbf{A}^m, \mathbf{D}^m) &= I(\delta^n; \mathbf{S}^m, \mathbf{D}^m) \\
&\leq H(\mathbf{S}^m, \mathbf{D}^m) \\
&= H(D_1 - S_1, \dots, D_m - S_m) \\
&\stackrel{(b)}{\leq} \sum_{k: S_k \text{ is odd}} H(D_k - S_k)
\end{aligned} \tag{37}$$

where (b) follows from the fact that the attacker is served with priority in even time slots, i.e.,  $D_k - S_k \equiv 1$ , where  $S_k$  is even. On the other hand, since the user is served first during the odd time slots, similar to (31) we have

$$D_k - S_k = \begin{cases} 1 & \text{if } q(S_k) = 0 \text{ and } \delta_{S_k} = 0, \\ 2 & \text{otherwise.} \end{cases} \tag{38}$$

Applying (38) to (37) and taking the limit at  $\lambda \rightarrow 0$ , we have

$$\begin{aligned}
\lim_{\lambda \rightarrow 0} I(\delta^n; \mathbf{A}^m, \mathbf{D}^m) &\leq \sum_{k: S_k \text{ is odd}} \lim_{\lambda \rightarrow 0} H(\mathbb{P}(q(S_k) = 0) \cdot (1 - \lambda)) \\
&\stackrel{(c)}{=} \sum_{k: S_k \text{ is odd}} H(\lambda),
\end{aligned} \tag{39}$$

where (c) holds because

$$\lim_{\lambda \rightarrow 0} \mathbb{P}(q(S_k) = 0) = 1, \quad k \in \mathbb{Z}. \tag{40}$$

See Appendix G for the proof of (40).

Plugging (40) into (39), we have

$$\begin{aligned}
\lim_{\lambda \rightarrow 0} \mathcal{R}_{WC-TDMA}(\lambda) &\leq \max_{\mathbf{S}: \lim_{k \rightarrow \infty} \frac{k}{S_k} < 1 - \lambda} \lim_{n \rightarrow \infty} \frac{\sum_{k: S_k \text{ is odd}} H(D_k - S_k)}{nH(\lambda)} \\
&\stackrel{(d)}{\leq} \frac{1}{2}.
\end{aligned} \tag{41}$$

where (d) holds because at most half of time slots are odd. (41) together with (36) prove (35). ■

#### E. Deterministic Work-Conserving Schedulers

In this section, we derive a universal lower bound on the information leakage for the class of deterministic work-conserving (det-WC) schedulers, where the scheduler's actions are deterministic (the same arrival instances result in the same departure events), and the scheduler can idle only if there are no jobs in the queue.

*Theorem 4.7:* The information leakage of a det-WC scheduler serving a user and an attacker is lower-bounded

by

$$\mathcal{L}_{det-WC}(\lambda) \geq \max_{\omega: \omega < 1-\lambda} \frac{\omega(z_0 - 1)}{2z_0} H(\lambda), \quad (42)$$

where  $\lambda$  is the user's arrival rate, and  $z_0$  is the only root of the equation

$$\alpha_\omega z(1 - \lambda + \lambda z)^{\lceil \frac{1}{\omega} \rceil} + (1 - \alpha_\omega) z^2(1 - \lambda + \lambda z)^{\lfloor \frac{1}{\omega} \rfloor} - z^{\lceil \frac{1}{\omega} \rceil} = 0, \quad (43)$$

outside the unit circle.

*Proof:* We again consider the periodic-sampling attack,  $\mathbf{A}^*$ , defined in (11), under which the mutual information between the attacker's observation and the user's arrival pattern is given by

$$\begin{aligned} I(\delta^n; \mathbf{A}^{*m}, \mathbf{D}^{*m}) &\stackrel{(a)}{=} H(\delta^n) - \sum_{i=1}^n H(\delta_i | \delta^{i-1}, \mathbf{A}^{*m}, \mathbf{D}^{*m}) \\ &\stackrel{(b)}{=} H(\delta^n) - \sum_{i=1}^n H(\delta_i | q(1), \dots, q(i), \mathbf{A}^{*m}, \mathbf{D}^{*m}) \\ &\geq H(\delta^n) - \sum_{i=1}^n H(\delta_i | q(i), \mathbf{A}^{*m}) \end{aligned} \quad (44)$$

where (a) follows from the entropy chain rule. (b) holds because given  $\delta^{i-1}$  and  $\mathbf{A}^{*m}$ , queue lengths at all time slots up to time  $i$  are known.

Define  $a_i$  as the indicator of the attacker's arrival event in time slot  $i$ . Following (11),  $a_i$ 's are i.i.d. distributed as

$$a_i = \begin{cases} 1 & \text{w.p. } \omega \\ 0 & \text{w.p. } 1 - \omega \end{cases}, \quad i \in \mathbb{Z}. \quad (45)$$

Applying (45) to (44), we have

$$I(\delta^n; \mathbf{A}^{*m}, \mathbf{D}^{*m}) \geq H(\delta^n) - \sum_{i=1}^n H(\delta_i | q(i), a_i) \quad (46)$$

At some time slot  $i$ , assume the scheduler serves the user first when a tie of queue length occurs. Additionally, assume the queue is empty, i.e.,  $q(i) = 0$ , and the attacker issues a new job, i.e.,  $a_i = 1$ . As argued before, in this case the attacker knows whether the user issued a job or not in this slot without ambiguity. Since the scheduler does not know the users' identities, in the worst case, the user has priority for at least half of time slots. Therefore, the afore mentioned scenario would at least occur with probability by  $\frac{1}{2}\omega\mathbb{P}(q(i) = 0)$ . Plugging this into (46), we have

$$\begin{aligned} I(\delta^n; \mathbf{A}^{*m}, \mathbf{D}^{*m}) &\geq H(\delta^n) - \sum_{i=1}^n \left(1 - \frac{1}{2}\omega\mathbb{P}(q(i) = 0)\right) H(\delta_i) \\ &= \frac{1}{2}\omega\mathbb{P}(q(i) = 0) H(\lambda). \end{aligned} \quad (47)$$



We next derive the probability that the attacker sees an empty queue,  $\mathbb{P}(q(i) = 0)$ . From (11), we can write the update equation for queue lengths sampled by the attacker as

$$q(A_{k+1}^*) = \left( q(A_{k+1}^*) + Y_k - \left\lceil \frac{1}{\omega} \right\rceil \right)_+, \quad k \in \mathbb{Z}, \quad (48)$$

where  $Y_k$ 's are i.i.d. distributed as

$$Y_k = \begin{cases} 2 + X_k^* & \text{w.p. } \alpha_\omega \\ 1 + X_k^{*'} & \text{w.p. } 1 - \alpha_\omega \end{cases}, \quad (49)$$

where  $X_k^* \sim \text{Binomial}(\lfloor \frac{1}{\omega} \rfloor, \lambda)$ ,  $X_k^{*'} \sim \text{Binomial}(\lceil \frac{1}{\omega} \rceil, \lambda)$ . The probability of the attacker seeing an empty queue can be derived by calculating  $z$ -transform of  $q(A_{k+1}^*)$  in the steady state [27, (3)], which is given by

$$\lim_{k \rightarrow \infty} \mathbb{P}(q(A_{k+1}^*) = 0) = \frac{z_0 - 1}{z_0}, \quad (50)$$

where  $z_0$  is the only root of  $\alpha_\omega z(1 - \lambda + \lambda z)^{\lceil \frac{1}{\omega} \rceil} + (1 - \alpha_\omega)z^2(1 - \lambda + \lambda z)^{\lfloor \frac{1}{\omega} \rfloor} - z^{\lceil \frac{1}{\omega} \rceil} = 0$  outside the unit circle.

(47) and (50) readily imply (42). ■

We plot the numerical solution of the bound in (42) in Figure 2. As can be seen, the attacker always gains a significant amount of information of the user in a det-WC scheduler, especially when the user's rate  $\lambda$  is below 0.5. More specifically, as  $\lambda \rightarrow 0$ , the user's traffic pattern is leaked for at least half of the time.

## V. CONCLUSION

Timing side channels in deterministic work-conserving schedulers were studied, where information leakage happens due to the sharing of queue among users. Our analysis proved that commonly deployed work-conserving queue service policies all leak significant amounts of user's information. If the scheduler adds idling slots, or randomizes the service order of jobs to some extent, the attacker may not be able to make accurate inferences about queue states any more and subsequently the user's arrival pattern. Such mitigation measures for the timing side channel of our interest remain open.

## APPENDIX

## A. Proof of Theorem 4.2

Before proving Theorem 4.2, we introduce two lemmas.

Define function  $\mathcal{H}_\lambda: \mathbb{Z}^+ \rightarrow \mathbb{R} \cup \{+\infty\}$  as

$$\mathcal{H}_\lambda(i) = H \left( \delta_1, \delta_2, \dots, \delta_i \middle| \sum_{j=1}^i \delta_j \right), \quad (51)$$

where  $\delta_i$ 's are i.i.d. Bernoulli random variables.

*Lemma A.1:*  $\mathcal{H}_\lambda(\cdot)$  is a mid-point convex function [28, (2.8)]; i.e.,

$$\mathcal{H}_\lambda(a) + \mathcal{H}_\lambda(b) \geq \mathcal{H}_\lambda \left( \left\lfloor \frac{a+b}{2} \right\rfloor \right) + \mathcal{H}_\lambda \left( \left\lceil \frac{a+b}{2} \right\rceil \right), \quad (52)$$

for all  $a \leq b$ ,  $a, b \in \mathbb{Z}^+$ , and the equality is achieved if  $b = a$  or  $b = a + 1$ .

*Proof:* Given  $m, n \in \mathbb{Z}^+$ ,  $m \geq n$ , we compute

$$\begin{aligned} \mathcal{H}_\lambda(m) - \mathcal{H}_\lambda(n) &= H \left( \delta^n, \delta_{n+1}^m \middle| \sum_{i=1}^m \delta_i \right) - H \left( \delta^n \middle| \sum_{i=1}^n \delta_i \right) \\ &\stackrel{(a)}{=} H \left( \delta_{n+1}^m \middle| \sum_{i=1}^m \delta_i \right) + H \left( \delta^n \middle| \delta_{n+1}^m, \sum_{i=1}^m \delta_i \right) - H \left( \delta^n \middle| \sum_{i=1}^n \delta_i \right) \\ &\stackrel{(b)}{=} H \left( \delta_{n+1}^m \middle| \sum_{i=1}^m \delta_i \right) \\ &\stackrel{(c)}{=} H \left( \delta^{m-n} \middle| \sum_{i=1}^m \delta_i \right) \end{aligned} \quad (53)$$

where (a) applies the entropy chain rule [24, Theorem 2.2.1], (b) holds because that  $\sum_{i=1}^n \delta_i$  is a sufficient statistic to infer  $\delta^n$  and can be calculated from  $\delta_{n+1}^m$  and  $\sum_{i=1}^m \delta_i$ , and (c) follows from the fact that Bernoulli arrivals are uniformly distributed once the total number of arrivals is known.

Replacing  $(m, n)$  with  $(b, \lceil \frac{a+b}{2} \rceil)$  and  $(\lfloor \frac{a+b}{2} \rfloor, a)$  in (53), respectively, we get

$$\mathcal{H}_\lambda(b) - \mathcal{H}_\lambda \left( \left\lceil \frac{a+b}{2} \right\rceil \right) = H \left( \delta^{\lfloor \frac{a+b}{2} \rfloor - a} \middle| \sum_{i=1}^b \delta_i \right) \quad (54)$$

and

$$\mathcal{H}_\lambda \left( \left\lfloor \frac{a+b}{2} \right\rfloor \right) - \mathcal{H}_\lambda(a) = H \left( \delta^{\lfloor \frac{a+b}{2} \rfloor - a} \middle| \sum_{i=1}^{\lfloor \frac{a+b}{2} \rfloor} \delta_i \right). \quad (55)$$

Subtracting (55) from (54), we get

$$\begin{aligned}
\mathcal{H}_\lambda(b) + \mathcal{H}_\lambda(a) - \mathcal{H}_\lambda\left(\left\lceil \frac{a+b}{2} \right\rceil\right) - \mathcal{H}_\lambda\left(\left\lfloor \frac{a+b}{2} \right\rfloor\right) &= H\left(\delta^{\lfloor \frac{a+b}{2} \rfloor - a} \middle| \sum_{i=1}^b \delta_i\right) - H\left(\delta^{\lfloor \frac{a+b}{2} \rfloor - a} \middle| \sum_{i=1}^{\lfloor \frac{a+b}{2} \rfloor} \delta_i\right) \\
&\stackrel{(d)}{=} H\left(\delta^{\lfloor \frac{a+b}{2} \rfloor - a} \middle| \sum_{i=1}^b \delta_i\right) - H\left(\delta^{\lfloor \frac{a+b}{2} \rfloor - a} \middle| \sum_{i=1}^{\lfloor \frac{a+b}{2} \rfloor} \delta_i, \sum_{i=1}^b \delta_i\right) \\
&= I\left(\delta^{\lfloor \frac{a+b}{2} \rfloor - a}, \sum_{i=1}^{\lfloor \frac{a+b}{2} \rfloor} \delta_i \middle| \sum_{i=1}^b \delta_i\right) \geq 0
\end{aligned} \tag{56}$$

where (d) follows from that fact that  $\sum_{i=\lfloor \frac{a+b}{2} \rfloor + 1}^b \delta_i$  does not provide extra information to infer  $\delta^{\lfloor \frac{a+b}{2} \rfloor - a}$ . Clearly, the last inequality turns equality if  $b = a$  or  $a + 1$ , which completes the proof. ■

**Lemma A.2:**  $\mathcal{H}_\lambda(\cdot)$  is an integrally-convex function [28, (2.5)]; i.e., it can be extended to a globally convex function  $\hat{\mathcal{H}}_\lambda: \mathbb{R}^+ \rightarrow \mathbb{R} \cup \{+\infty\}$ , where

$$\hat{\mathcal{H}}_\lambda(x) = \alpha_x \mathcal{H}_\lambda(\lfloor x \rfloor) + (1 - \alpha_x) \mathcal{H}_\lambda(\lceil x \rceil) \tag{57}$$

where  $\alpha_x = \frac{\lceil \frac{1}{x} \rceil - \frac{1}{x}}{\lceil \frac{1}{x} \rceil - \lfloor \frac{1}{x} \rfloor}$ .

*Proof:* Applying Lemma A.1, this is a direct result from [28, Theorem 2.4], which states that a discrete function satisfying mid-point convexity can be extended to a convex continuous function through linear interpolation. ■

**Proof of Theorem 4.2:** The entropy of the sampled process satisfies

$$\begin{aligned}
H(N(A_1), N(A_2), \dots, N(A_k)) &\stackrel{(a)}{=} I(\delta^{A_k-1}; N(A_1), N(A_2), \dots, N(A_k)) \\
&= I(\delta^{A_k-1}; N(A_2) - N(A_1), \dots, N(A_k) - N(A_{k-1})) \\
&\stackrel{(b)}{=} H(\delta^{A_k-1}) - \sum_{i=1}^{k-1} H(\delta_{A_i}^{A_{i+1}-1} | N(A_{i+1}) - N(A_i)),
\end{aligned} \tag{58}$$

where (a) follows from that  $N(A_i)$ 's are functions of  $\delta$  and (b) applies the entropy chain rule.

Define  $n_r$  to be the number of elements in the sequence  $\{A_2 - A_1, \dots, A_k - A_{k-1}\}$  that take value of  $r$ , then  $\sum_{r=1}^{\infty} n_r = k$ . The conditional entropy in (58) can be rewritten as

$$\begin{aligned}
\sum_{i=1}^{k-1} H(\delta_{A_i}^{A_{i+1}-1} | N(A_{i+1}) - N(A_i)) &= \sum_{r=1}^{\infty} n_r H\left(\delta^r \middle| \sum_{j=1}^r \delta_j\right) \\
&= \sum_{r=1}^{\infty} n_r \mathcal{H}_\lambda(r) \\
&\stackrel{(c)}{=} \sum_{r=1}^{\infty} n_r \hat{\mathcal{H}}_\lambda(r),
\end{aligned} \tag{59}$$

where function  $\mathcal{H}_\lambda(\cdot)$  and  $\hat{\mathcal{H}}_\lambda(\cdot)$  are defined according to (51) and (57), respectively, and (c) holds because  $\mathcal{H}_\lambda(\cdot)$  and  $\hat{\mathcal{H}}_\lambda(\cdot)$  take the same values for integer arguments.

Combining (58) and (59), the entropy rate of the sampled process is upper-bounded as given by

$$\begin{aligned}
\lim_{k \rightarrow \infty} \frac{H(N(A_1), \dots, N(A_k))}{k} &= \omega H(\delta_1) - \lim_{k \rightarrow \infty} \frac{\sum_{r=1}^{\infty} n_r \hat{\mathcal{H}}_\lambda(r)}{k} \\
&\stackrel{(d)}{\leq} \omega H(\delta_1) - \lim_{k \rightarrow \infty} \hat{\mathcal{H}}_\lambda \left( \frac{\sum_{r=1}^{\infty} n_r r}{k} \right) \\
&\leq \omega H(\delta_1) - \hat{\mathcal{H}}_\lambda \left( \lim_{k \rightarrow \infty} \frac{\sum_{r=1}^{\infty} n_r r}{k} \right) \\
&\stackrel{(e)}{=} \omega H(\delta_1) - \hat{\mathcal{H}}_\lambda \left( \frac{1}{\omega} \right) \\
&\stackrel{(f)}{=} \alpha_\omega H \left( \sum_{i=1}^{\lfloor \frac{1}{\omega} \rfloor} \delta_i \right) + (1 - \alpha_\omega) H \left( \sum_{i=1}^{\lceil \frac{1}{\omega} \rceil} \delta_i \right),
\end{aligned} \tag{60}$$

where (d) applies Jensen's inequality [29, (9.1.3.1)], (e) follows from  $\omega = \lim_{k \rightarrow \infty} \frac{k}{A_k}$ , and (f) follows from (51) and (57).

Finally, it is easy to verify that the sampled process by the periodic-sampling strategy has the same entropy rate as given by the bound in (60), which completes the proof for (12).

### B. Proof of Achievability in Theorem 4.3

For the achievability of (13), we need to show

$$\mathcal{L}_{FCFS}(\lambda) \geq (1 - \lambda) \left( \alpha_{1-\lambda} H \left( \sum_{i=1}^{\lfloor \frac{1}{1-\lambda} \rfloor} \delta_i \right) + (1 - \alpha_{1-\lambda}) H \left( \sum_{i=1}^{\lceil \frac{1}{1-\lambda} \rceil} \delta_i \right) \right).$$

*Proof:* Let  $\mathbf{D}^* = \{D_1^*, D_2^*, \dots\}$  denote the departure times of the attacker's jobs applying the periodic sampling attack strategy defined in (11). We have a lower bound on the information leakage as

$$\mathcal{L}_{FCFS}(\lambda) \geq \max_{\mathbf{A}^*; \lim_{k \rightarrow \infty} \frac{k}{A_k^*} < 1-\lambda} \lim_{n \rightarrow \infty} \frac{I(\boldsymbol{\delta}^n; \mathbf{A}^{*m}, \mathbf{D}^{*m})}{n}. \tag{61}$$

Rewrite the mutual information in (61) as follows:

$$\begin{aligned}
I(\boldsymbol{\delta}^n; \mathbf{A}^{*m}, \mathbf{D}^{*m}) &\stackrel{(a)}{=} I(\boldsymbol{\delta}^n; \mathbf{A}^{*m}, q(A_1^*), q(A_2^*), \dots, q(A_m^*)) \\
&\stackrel{(b)}{=} I(\boldsymbol{\delta}^n; \mathbf{A}^{*m}, \mathbf{X}^{*m}) - I(\boldsymbol{\delta}^n; \mathbf{A}^{*m}, \mathbf{X}^{*m} | \mathbf{A}^{*m}, q(A_1^*), \dots, q(A_m^*)) \\
&= I(\boldsymbol{\delta}^n; \mathbf{A}^{*m}, \mathbf{X}^{*m}) - H(\mathbf{X}^{*m} | \mathbf{A}^{*m}, q(A_1^*), \dots, q(A_m^*)) \\
&\stackrel{(c)}{=} H(\mathbf{X}^{*m} | \mathbf{A}^{*m}) - H(\mathbf{X}^{*m} | \mathbf{A}^{*m}, q(A_1^*), \dots, q(A_m^*)),
\end{aligned} \tag{62}$$

where (a) follows from (8),  $X_k^*$  is the total number of the user's jobs that have arrived between the times  $A_{k-1}^*$

and  $A_k^*$ , (b) follows from the Markov chain in (16), and (c) holds because  $\mathbf{A}^{*m}$  is independent of  $\delta^n$ .

Substituting (62) into (61), we have

$$\mathcal{L}_{FCFS}(\lambda) \geq \max_{\omega: \omega < 1-\lambda} \omega \lim_{m \rightarrow \infty} \frac{H(\mathbf{X}^{*m} | \mathbf{A}^{*m}) - H(\mathbf{X}^{*m} | \mathbf{A}^{*m}, q(A_1^*), \dots, q(A_m^*))}{m}. \quad (63)$$

Applying the entropy chain rule to the second term in (63), we have

$$\begin{aligned} \lim_{m \rightarrow \infty} \frac{H(\mathbf{X}^{*m} | \mathbf{A}^{*m}, q(A_1^*), \dots, q(A_m^*))}{m} &= \lim_{m \rightarrow \infty} \frac{\sum_{k=1}^m H(X_k^* | \mathbf{X}^{*k-1}, \mathbf{A}^{*m}, q(A_1^*), \dots, q(A_m^*))}{m} \\ &\stackrel{(d)}{=} \lim_{m \rightarrow \infty} \frac{\sum_{k=1}^m H(X_k^* | A_k^* - A_{k-1}^*, q(A_{k-1}^*), q(A_k^*))}{m}, \end{aligned} \quad (64)$$

where (d) follows from the update equation of queue length seen by the attacker's jobs, which is given by

$$q(A_k^*) = (q(A_{k-1}^*) + 1 + X_k^* - (A_k^* - A_{k-1}^*))_+. \quad (65)$$

It can be shown that  $\{A_k^* - A_{k-1}^*, q(A_{k-1}^*), q(A_k^*)\}$ ,  $k \in \mathbb{Z}$ , forms a positive recurrent Markov chain (See Appendix D for the proof), which implies the equivocation rate in (64) converges as  $k \rightarrow \infty$ , with the limit determined by the stationary distribution of  $\{A_k^* - A_{k-1}^*, q(A_{k-1}^*), q(A_k^*)\}$ . Let  $\{\mathcal{T}, Q_1, Q_2\}$  take the stationary distribution of  $\{A_k^* - A_{k-1}^*, q(A_{k-1}^*), q(A_k^*)\}$ . From Cesàro mean theorem [24, Theorem 4.2.3], (64) can be rewritten as

$$\lim_{m \rightarrow \infty} \frac{H(\mathbf{X}^{*m} | \mathbf{A}^{*m}, q(A_1^*), \dots, q(A_m^*))}{m} = H(\mathcal{X} | \mathcal{T}, Q_1, Q_2). \quad (66)$$

Furthermore, it can be shown that as  $\omega \rightarrow 1 - \lambda$ ,  $Q_2$  is always positive (See Appendix C for the proof), i.e.,

$$\lim_{\omega \rightarrow 1-\lambda} \mathbb{P}(Q_2 = 0) = 0. \quad (67)$$

From the queue length update equation,  $Q_2 = (Q_1 + 1 + \mathcal{X} - \mathcal{T})_+$ , we have that

$$H(\mathcal{X} | \mathcal{T}, Q_1, Q_2 > 0) = 0. \quad (68)$$

(67) and (68) imply that

$$\lim_{\omega \rightarrow 1-\lambda} H(\mathcal{X} | \mathcal{T}, Q_1, Q_2) = 0. \quad (69)$$

Substituting (66) into (63) and applying (69), we have

$$\begin{aligned}
\mathcal{L}_{FCFS}(\lambda) &\geq \max_{\omega: \omega < 1-\lambda} \omega \left( \lim_{m \rightarrow \infty} \frac{H(\mathbf{X}^{*m} | \mathbf{A}^{*m})}{m} - H(\mathcal{X} | \mathcal{T}, Q_1, Q_2) \right) \\
&\geq \lim_{\omega \rightarrow 1-\lambda} \omega \left( \lim_{m \rightarrow \infty} \frac{H(\mathbf{X}^{*m} | \mathbf{A}^{*m})}{m} - H(\mathcal{X} | \mathcal{T}, Q_1, Q_2) \right) \\
&= \lim_{\omega \rightarrow 1-\lambda} \omega \lim_{m \rightarrow \infty} \frac{H(\mathbf{X}^{*m} | \mathbf{A}^{*m})}{m}.
\end{aligned} \tag{70}$$

Since  $X_i^*$ 's are i.i.d.  $\text{Binomial}(A_i^* - A_{i-1}^*, \lambda)$  random variables, we have

$$\lim_{m \rightarrow \infty} \frac{H(\mathbf{X}^{*m} | \mathbf{A}^{*m})}{m} = \alpha_\omega H\left(\sum_{i=1}^{\lfloor \frac{1}{\omega} \rfloor} \delta_i\right) + (1 - \alpha_\omega) H\left(\sum_{i=1}^{\lceil \frac{1}{\omega} \rceil} \delta_i\right), \tag{71}$$

where  $\alpha_\omega = \frac{\lceil \frac{1}{\omega} \rceil - \frac{1}{\omega}}{\lceil \frac{1}{\omega} \rceil - \lfloor \frac{1}{\omega} \rfloor}$ . Substituting (71) into (70) proves (61).  $\blacksquare$

### C. Proof of (67)

Recall that  $\mathcal{T}$  is distributed as (11),  $\mathcal{X}$  is  $\text{Binomial}(\mathcal{T}, \lambda)$ , and  $Q_1$  and  $Q_2$  have identical distribution satisfying

$$Q_2 = (Q_1 + 1 + \mathcal{X} - \mathcal{T})_+. \tag{72}$$

We need to prove that  $\lim_{\omega \rightarrow 1-\lambda} \mathbb{P}(Q_2 = 0) = 0$ .

*Proof:* First, (72) can be rewritten as

$$Q_2 = \left( Q_1 + Y - \left\lceil \frac{1}{\omega} \right\rceil \right)_+, \tag{73}$$

where

$$Y = \begin{cases} 2 + X_1 & \text{w.p. } \alpha_\omega \\ 1 + X'_1 & \text{w.p. } 1 - \alpha_\omega \end{cases}, \tag{74}$$

where  $X_1 \sim \text{Binomial}(\lfloor \frac{1}{\omega} \rfloor, \lambda)$ ,  $X'_1 \sim \text{Binomial}(\lceil \frac{1}{\omega} \rceil, \lambda)$ , and  $\alpha_\omega = \frac{\lceil \frac{1}{\omega} \rceil - \frac{1}{\omega}}{\lceil \frac{1}{\omega} \rceil - \lfloor \frac{1}{\omega} \rfloor}$ .

Taking the  $z$ -transform of  $Q_2$ , we have

$$Q(z) = \frac{\sum_{k=0}^{\lfloor \frac{1}{\omega} \rfloor - 2} \sum_{r=0}^{\lfloor \frac{1}{\omega} \rfloor - 2 - k} p_k u_r \left( z^{\lceil \frac{1}{\omega} \rceil - 1} - z^{k+r} \right)}{z^{\lceil \frac{1}{\omega} \rceil - 1} - \mathcal{Y}(z)} \tag{75}$$

where  $\mathcal{Y}(z)$  is the  $z$ -transform of  $Y$  as given by

$$\mathcal{Y}(z) = \alpha_\omega z(1 - \lambda + \lambda z)^{\lceil \frac{1}{\omega} \rceil} + (1 - \alpha_\omega) z^2(1 - \lambda + \lambda z)^{\lfloor \frac{1}{\omega} \rfloor}, \tag{76}$$

and  $p_k = \mathbb{P}(Q_2 = k)$  and  $u_r = \mathbb{P}(Y = r)$ .

Substituting (76) into (75) and letting  $z = 1$  on both sides, we have

$$\sum_{k=0}^{\lceil \frac{1}{\omega} \rceil - 2} \left( \sum_{r=0}^{\lceil \frac{1}{\omega} \rceil - 2 - r} u_r \left( \left\lceil \frac{1}{\omega} \right\rceil - 1 - k - r \right) \right) = \left\lceil \frac{1}{\omega} \right\rceil (1 - \omega - \lambda). \quad (77)$$

Dropping the terms with  $r > 1$  on the left hand side, we further get

$$u_1 \left( \left\lceil \frac{1}{\omega} \right\rceil - 1 - k \right) \sum_{k=0}^{\lceil \frac{1}{\omega} \rceil - 2} p_k \leq \left\lceil \frac{1}{\omega} \right\rceil (1 - \omega - \lambda). \quad (78)$$

Plugging in the values of  $u_1$  and taking the limit as  $\omega \rightarrow 1 - \lambda$ , we have

$$\lim_{\omega \rightarrow 1 - \lambda} \sum_{k=0}^{\lceil \frac{1}{\omega} \rceil - 2} p_k \leq \lim_{\omega \rightarrow 1 - \lambda} \frac{\left\lceil \frac{1}{\omega} \right\rceil (1 - \omega - \lambda)}{(1 - \alpha_\omega) (1 - \lambda \lceil \frac{1}{\omega} \rceil)} = 0, \quad (79)$$

which readily implies  $p_0 = 0$ , which is the desired result.  $\blacksquare$

#### D. Queuing Analysis of the FCFS scheduler Under a Periodic Sampling Attack

*Theorem A.3:* In an FCFS scheduler with total job arrival rate below 1, when the attacker applies the periodic-sampling strategy defined in (11), the tuples  $\{A_k^* - A_{k-1}^*, q(A_{k-1}^*), q(A_k^*)\}$ ,  $k \in \mathbb{Z}$ , form a positive recurrent Markov chain.

*Proof:* We first prove that  $\{q(A_k^*)\}$ ,  $k \in \mathbb{Z}$ , form a positive recurrent Markov chain. The Markovian property directly follows from the FCFS policy and memoryless property of the user's arrival process; given the queue length at  $A_k^*$ , future queue states are independent with the past arrival history.

We show the ergodicity using a linear Lyapunov function defined as

$$V(q(A_k^*)) = q(A_k^*), k \in \mathbb{Z}. \quad (80)$$

Recall that the arrival rates of the user and attacker by  $\lambda$  and  $\omega$ , respectively. If  $q(A_k^*) \geq \lceil \frac{1}{\omega} \rceil$ , the scheduler is guaranteed to be busy from  $A_k^*$  to  $A_{k+1}^*$ . Hence, from (65),

$$q(A_{k+1}^*) = q(A_k^*) + 1 + X_k^* - (A_{k+1}^* - A_k^*). \quad (81)$$

As  $X_k^*$  has mean as  $\frac{\lambda}{\omega}$ , and  $A_{k+1}^* - A_k^*$  has mean as  $\frac{1}{\omega}$ , the drift of the Lyapunov function in this case is given by

$$\mathbb{P}V(q(A_k^*)) - V(q(A_k^*)) = -\frac{1 - \omega - \lambda}{\omega}. \quad (82)$$

Additionally, during  $[A_k^*, A_{k+1}^*)$ , the buffer queue length can grow at most 1, so the drift is bounded by

$$\mathbb{P}V(q(A_k^*)) - V(q(A_k^*)) \leq 1. \quad (83)$$

Combining (82) and (83), the drift in any queue state is bound by

$$\mathbb{P}V(q(A_k^*)) - V(q(A_k^*)) \leq -\epsilon + \mathbb{1}_{\{q(A_k^*) < \lceil \frac{1}{\omega} \rceil\}}, \quad (84)$$

where  $\epsilon = \frac{1-\omega-\lambda}{\omega}$ . Following Foster-Lyapunov stability [30, Theorem 5], (84) implies the Markov chain  $\{q(A_k^*)\}$ ,  $k \in \mathbb{Z}$ , is positive recurrent.

For the same reason we argue for the Markovian property of chain  $\{q(A_k^*)\}$ ,  $k \in \mathbb{Z}$ ,  $\{A_k^* - A_{k-1}^*, q(A_{k-1}^*), q(A_k^*)\}$ ,  $k \in \mathbb{Z}$ , also form a Markov chain. Additionally, a stationary state distribution of  $\{A_k^* - A_{k-1}^*, q(A_{k-1}^*), q(A_k^*)\}$ ,  $k \in \mathbb{Z}$ , can be derived from the stationary state distribution of  $\{q(A_k^*)\}$  and (11). The existence of the stationary distribution implies that  $\{A_k^* - A_{k-1}^*, q(A_{k-1}^*), q(A_k^*)\}$ ,  $k \in \mathbb{Z}$ , must be positive recurrent [31, Definition 3.1]. ■

#### E. Busy Period Distribution of the Round Robin Scheduler

Consider a round robin scheduler serving a user and an attacker. The user sends jobs according to a Bernoulli arrival process with rate  $\lambda \leq 0.5$ , and the attacker applies the nonstop monitoring attack, where the arrival and departure times satisfy (5). We prove the busy periods seen by the attacker,  $B_r$ 's as defined in (23), are i.i.d. distributed as (25) and have mean

$$\mathbb{E}[B_r] = \lim_{n \rightarrow \infty} \frac{\sum_{r=1}^n B_r}{n} = \frac{1}{1-2\lambda}. \quad (85)$$

*Proof:* Write the update equation of queue lengths seen by the attacker as

$$q(A_{k+1}) = \left( q(A_k) + 1 + \sum_{i=A_k+1}^{A_{k+1}} \delta_i - (A_{k+1} - A_k) \right)_+, \quad (86)$$

for  $k \in \mathbb{Z}$ .

From (5) and (22), we have

$$A_{k+1} - A_k = \begin{cases} 1 & \text{if } q(A_k) = 0 \\ 2 & \text{if } q(A_k) > 0 \end{cases}. \quad (87)$$

Given (86) and (87), we can draw a Markov chain formed by  $\{q(A_k)\}$ ,  $k \in \mathbb{Z}$ , as depicted by Figure 6. The length of busy period  $B_r$  is simply a function of the number of transitions,  $s$ , it takes to return back to state 0 (starting from state 0), as given by

$$B_r = 2s - 1, \quad r \in \mathbb{Z}. \quad (88)$$

Clearly,  $s$  has the same PMF as  $B_r$ 's.

From the Markov chain in Figure 6, we know

$$\mathbb{P}(s = 1) = 1 - \lambda. \quad (89)$$



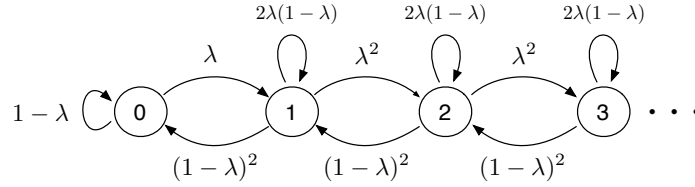


Fig. 6. The Markov chain of queue length when attacking the round robin scheduler with the nonstop monitoring strategy.

For  $s = 2$ , the queue length first needs one step to jump to state 1, and then returns to state 0, which has the probability as

$$\mathbb{P}(s = 2) = \lambda(1 - \lambda)^2. \quad (90)$$

For  $s > 2$ , after jumping to state 1 for first time, the queue state needs to experience another  $s - 2$  transitions before returning to state 1 for the last time and eventually returning back to state 0. Notice that each state transition either increases the queue length by 1, decreases the queue length by 1, or keeps the queue length unchanged. Therefore, the  $s - 2$  intermediate transitions must contain the same number of transitions that increase and decrease the queue length. Moreover, the increase in the queue length must always lead the decrease; otherwise, the queue would return to state 0 before the require number of transitions finish. Define  $W_{s,j}$  to be the total ways of  $s - 2$  transitions with  $j$  transitions increasing the queue (also  $j$  transitions decreasing the queue). Then  $W_{s,j} = \binom{s-2}{2j} C_j$ , where  $C_j$  is the famous Catalan number as given by

$$C_j = \frac{1}{j+1} \binom{2j}{j}. \quad (91)$$

The transition probability for  $s > 2$  can be calculated as

$$\begin{aligned} \mathbb{P}(s = r) &= \lambda(1 - \lambda)^2 \sum_{j=1}^{\lfloor \frac{r-2}{2} \rfloor} W_{r,j} (2(1 - \lambda)\lambda)^{r-2-2j} (1 - \lambda)^{2j} \lambda^{2j} \\ &= 2^{r-1} \lambda^{r-1} (1 - \lambda)^r \sum_{j=1}^{\lfloor \frac{r-2}{2} \rfloor} \frac{(r-2)! 2^{-2j-1}}{(r-2-2j)! j! (j+1)!}. \end{aligned} \quad (92)$$

(88), (89), (90), and (92) together imply (25).

In each busy period, the number of queue state transitions equals to the number of attacker's job arrivals. Since the attacker's arrival rate in this nonstop-monitoring attack is  $1 - \lambda$ , thus  $\mathbb{E}[\frac{s}{B_r}] = 1 - \lambda$ , which together with (88) prove (85). ■

#### F. Busy Period Distribution of the WC-TDMA Scheduler

Consider a WC-TDMA scheduler serving a user and an attacker, where even and odd time slots are reserved to the attacker and user, respectively. The user sends jobs according to Bernoulli arrival process with rate  $\lambda \leq 0.5$ ,

and the attacker sends jobs on all odd slots according to (30). We prove the busy periods seen by the attacker,  $B'_r$ 's as defined in (32), are i.i.d. distributed as (25) and have mean

$$\mathbb{E}[B'_r] = \frac{2 - 2\lambda}{1 - 2\lambda}, \quad \text{for } r \in \mathbb{Z}. \quad (93)$$

*Proof:* Based on (30) and (86), the evolving of queue state,  $\{q(A_k)\}$ ,  $k \in \mathbb{Z}$ , can be described by the same Markov chain in Figure 6, and the busy period  $B'_i$  is nothing but a function of the number of transitions,  $s$ , it takes to return to state 0 starting from state 0, as given by

$$B'_r = 2s, \quad r \in \mathbb{Z}. \quad (94)$$

Using the same arguments in our proof of (25) in Appendix E, it is clear that  $B'_r$ 's are distributed as (25). Additionally, from (88), (94), and (85),

$$\mathbb{E}[B'_r] = \mathbb{E}[B_r] + 1 = \frac{2 - 2\lambda}{1 - 2\lambda}. \quad (95)$$

■

#### G. Proof of (40)

We need to prove that

$$\lim_{\lambda \rightarrow 0} \mathbb{P}(q(S_k) = 0) = 1, \quad k \in \mathbb{Z}, \quad (96)$$

where  $S_k = \max\{A_k, D_{k-1}\}$ .

*Proof:* We prove this by induction. The base case is straightforward, considering the queue length starts with zero, i.e.,  $q(S_1) = 0$ . Now assume it is true that  $\lim_{\lambda \rightarrow 0} \mathbb{P}(q(S_k) = 0) = 1$ , for  $k = 1, 2, \dots, r$ . Consider the update equation of queue length, as given by

$$q(S_{r+1}) = \left( q(S_r) + 1 + \sum_{i=S_r}^{S_{r+1}-1} \delta_i - (S_{r+1} - S_r) \right)_+, \quad (97)$$

from which we calculate the probability of empty queue as given by

$$\begin{aligned}
& \lim_{\lambda \rightarrow 0} \mathbb{P}(q(S_{r+1}) = 0) \\
&= \lim_{\lambda \rightarrow 0} \sum_{i=0}^{S_{r+1}-S_r-1} \mathbb{P}(q(S_r) = i) \cdot \sum_{j=0}^{S_{r+1}-S_r-1-i} \binom{S_{r+1}-S_r}{j} (1-\lambda)^{S_{r+1}-S_r-j} \lambda^j \\
&\stackrel{(a)}{=} \lim_{\lambda \rightarrow 0} \sum_{j=0}^{S_{r+1}-S_r-1} \binom{S_{r+1}-S_r}{j} (1-\lambda)^{S_{r+1}-S_r-j} \lambda^j \\
&= 1 - \lim_{\lambda \rightarrow 0} \lambda^{S_{r+1}-S_r} \\
&= 1,
\end{aligned} \tag{98}$$

where (a) follows from the assumption that  $\lim_{\lambda \rightarrow 0} \mathbb{P}(q(S_k) = 0) = 1$ . This completes the proof.  $\blacksquare$

## REFERENCES

- [1] B. W. Lampson, “A note on the confinement problem,” *Commun. ACM*, vol. 16, no. 10, pp. 613–615, October 1973.
- [2] J. K. Millen, “Finite-state noiseless covert channels,” in *Computer Security Foundations Workshop*, Franconia, NH, 1989, pp. 81–86.
- [3] S. Cabuk, C. E. Brodley, and C. Shields, “IP covert channel detection,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 12, no. 4, 2009.
- [4] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds,” in *ACM Conf. on Computer and Communications Security (CCS)*, Chicago, IL, 2009, pp. 199–212.
- [5] X. Gong, N. Borisov, N. Kiyavash, and N. Schear, “Website detection using remote traffic analysis,” in *Privacy Enhancing Technologies (PETs)*, Vigo, Spain, 2012, pp. 58–78.
- [6] S. Kadloor and N. Kiyavash, “Delay optimal policies offer very little privacy,” in *IEEE International Conf. on Computer Communications (Infocom)*, Turin, Italy, 2013, pp. 2454–2462.
- [7] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Elsevier, 2007.
- [8] V. Anantharam and S. Verdú, “Bits through queues,” *IEEE Trans. on Inf. Theory*, vol. 42, no. 1, pp. 4–18, 1996.
- [9] S. H. Sellke, C.-C. Wang, N. Shroff, and S. Bagchi, “Capacity bounds on timing channels with bounded service times,” in *IEEE International Symposium on Inf. Theory*, Nice, France, 2007, pp. 981–985.
- [10] T. J. Riedl, T. P. Coleman, and A. C. Singer, “Finite block-length achievable rates for queuing timing channels,” in *IEEE Information Theory Workshop (ITW)*, Paraty, Brazil, 2011, pp. 200–204.
- [11] S. Gorantla, S. Kadloor, T. Coleman, N. Kiyavash, I. Moskowitz, and M. Kang, “Characterizing the efficacy of the NRL network pump in mitigating covert timing channels,” *IEEE Trans. on Inf. Forensics and Security*, vol. 7, no. 1, pp. 64–75, 2012.
- [12] J. Giles and B. Hajek, “An information-theoretic and game-theoretic study of timing channels,” *IEEE Trans. on Inf. Theory*, vol. 48, pp. 2455–2477, 2002.
- [13] A. Askarov, D. Zhang, and A. C. Myers, “Predictive black-box mitigation of timing channels,” in *ACM conference on Computer and communications security*, Chicago, IL, 2010, pp. 297–307.
- [14] D. Zhang, A. Askarov, and A. C. Myers, “Predictive mitigation of timing channels in interactive systems,” in *ACM conference on Computer and communications security*, Chicago, IL, 2011, pp. 563–574.
- [15] S. Murdoch and G. Danezis, “Low-cost traffic analysis of Tor,” in *IEEE Symposium on Security and Privacy*, V. Paxson and M. Waidner, Eds., Berkeley, CA, May 2005, pp. 183–195.

- [16] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *USENIX Security Symposium*, M. Blaze, Ed., San Diego, CA, 2004, pp. 303–320.
- [17] M. Rennhard and B. Plattner, "Introducing MorphMix: peer-to-peer based anonymous internet usage with collusion detection," in *ACM Workshop on Privacy in Electronic Society*, New York, NY, 2002, pp. 91–102.
- [18] N. Kiyavash, F. Koushanfar, T. P. Coleman, and M. Rodrigues, "A timing channel spyware for the CSMA/CA protocol," *IEEE Trans. on Inf. Forensics and Security*, vol. 8, no. 3, pp. 477–487, 2013.
- [19] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N. Borisov, "Low-cost side channel remote traffic analysis attack in packet networks," in *IEEE International Conference on Communications*, C. Xiao and J. C. Olivier, Eds., Cape Town, South Africa, 2010.
- [20] X. Gong, N. Kiyavash, and P. Venkitasubramaniam, "Information theoretic analysis of side channel information leakage in fcfs schedulers," in *IEEE International Symposium on Information Theory (ISIT)*, Saint-Petersburg, Russia, 2011, pp. 1255–1259.
- [21] A. D. Wyner, "The wire-tap channel," *Bell Sys. Tech. J.*, vol. 54, pp. 1355–1387, 1975.
- [22] S. Shakkottai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *Translations of the American Mathematical Society-Series 2*, vol. 207, pp. 185–202, 2002.
- [23] H. R. Gail, G. Grover, R. Guérin, S. L. Hantler, Z. Rosberg, and M. Sidi, "Buffer size requirements under longest queue first," *Performance Evaluation*, vol. 18, no. 2, pp. 133–140, 1993.
- [24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1987.
- [25] E. L. Hahne and R. G. Gallager, "Round robin scheduling for fair flow control in data communication networks," *NASA STI/Recon Technical Report N*, vol. 86, p. 30047, 1986.
- [26] L. Siegele, *Let it rise: A special report on corporate IT*. Economist Newspaper, 2008.
- [27] H. Bruneel, "Message delay in tdma channels with contiguous output," *IEEE Trans. on Communication*, vol. 34, no. 7, pp. 681–684, 1986.
- [28] K. Murota and A. Shioura, "Relationship of M-/L-convex functions with discrete convex functions by Miller and Favati-Tardella," *Discrete Applied Mathematics*, vol. 115, pp. 151–176, 2001.
- [29] S. G. Krantz, *Handbook of Complex Variables*. MA: Birkhäuser, 1995.
- [30] F. G. Foster, "On the stochastic matrices associated with certain queuing processes," *Ann. Math. Statistics*, vol. 24, pp. 355–360, 1953.
- [31] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1995.